DTIC FILE COPY

AD-A229 633

NWC TP 7013

Equations of Learning and Capacity of Layered Neural Networks

by
Jorge M. Martin
Applied Mathematics Research Group
Research Department

MAY 1989

NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555-6001

Approved for public release, distribution is unlimited.

DTIC
ELECTE
DEC 27 1990
S B D

050

Naval Weapons Center

FOREWORD

The purpose of this report is to document some of the basic results on the mathematical foundations of multilayered neural networks that were discovered during the initial phase of the Independent Research project entitled "The Mathematics of Artificial Neural Systems." These results do not represent mathematical breakthroughs per se. They were arrived at by applying well-known mathematical results and represent a small contribution to the basic knowledge of the mathematical aspects of the newly emergent theory of Artificial Neural Networks. This report may serve as a basis for subsequent publications and, to a limited extent, it may serve as a guide for design considerations and applications. The work was done at the Naval Weapons Center from October 1987 to May 1989.

This report has been reviewed for technical accuracy by W. O. Alltop.

Approved by
R. L. DERR, *Head*
Research Department
15 May 1989

Under authority of
J. A. BURT
Capt., U.S. Navy
Commander

Released for publication by
G. R. SCHIEFER
Technical Director

NWC Technical Publication 7013

Published by ..... Technical Information Department
Collation ..... Cover, 19 leaves
First printing ..... 75 copies

UNCLASSIFIED
ECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT A Statement; public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NWC TP 7013	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Weapons Center 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) China Lake, CA 93555-6001	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING SPONSORING ORGANIZATION Naval Weapons Center 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) China Lake, CA 93555-6001	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 61152N; PROJECT NO:; TASK NO: R00N00; WORK UNIT ACCESSION NO: 13807004

11 TITLE (Include Security Classification)

EQUATIONS OF LEARNING AND CAPACITY OF LAYERED NEURAL NETWORKS

12 PERSONAL AUTHOR(S)
Martin, J.M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM Oct 87 TO May 89	1989, May	35

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Layered Neural Networks, Transfer Function, Equations of Learning, Degrees of Freedom, Dimension, Capacity, Architecture.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Learning in a layered neural network (LNN) amounts to finding the correct interconnection weights that will produce the right input-output pairs. The input-output pairs and the architecture of the net define equations that the weights must satisfy. These equations, the Equations of Learning, are derived in this paper. By applying well-known results from dimension theory to these equations, one can derive an upper bound on the number of different input-output pairs that a layered network can learn.

Two simple examples are used to illustrate this result and its limitations. While analyzing the first example, it was discovered that the saturation of the sigmoid function is a desirable feature. The concepts of 'architecture' and 'capacity' of an LNN are defined, and a few results on architectures with maximal capacity are included. It was found that reducing the dimension of the output patterns increases the capacity of the LNN.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Jorge M. Martin	22b TELEPHONE (Include Area Code) 619-939-3034 22c OFFICE SYMBOL 3807

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—607-044

CONTENTS

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

QUALITY INSPECTED 1

ACKNOWLEDGMENT

The author is grateful to William Alltop for helpful discussions.

1. INTRODUCTION

Several neurocomputing paradigms, generally known as artificial neural networks, have been proposed in recent years. Some of the most popular are the Hopfield Net (References 1 and 2), the Adaptive Resonance Theory of Grossberg and Carpenter (References 3 through 6), the Adaptive Linear Combiner (ADALINE, MADALINE) of Widrow (References 7 and 8), Rumelhart's Multilayered Neural Networks (References 9 and 10), and variations of these. The associative memory capacity of Hopfield-type nets has been analyzed by several authors (References 11 through 15); the capacity of multilevel threshold functions was investigated (Reference 16). In this paper, we study the capacity of multilayered neural networks.

An upper bound on the number of patterns (input-output pairs) that a layered neural network can learn is derived. The result is obtained by applying some results from dimension theory to a set of equations that the input-output pairs must satisfy for the given architecture. These equations are interesting by themselves. They have a dual interpretation. For a given set of interconnection weights, the equations represent the transfer function (TF) between input and output patterns. Thus, if the net is to have a desired TF—that is, a desired set of input-output pairs of patterns—then the interconnection weights must satisfy the TF equations. In this sense, the equations represent equations of learning.

In Section 2, the general architecture of our layered neural networks (LNNs) will be presented together with the notation that we shall use to represent the general TF in closed form. In Section 3, we shall interpret these equations as defining a mapping from weight-space to output-space, at which

point the result on the capacity of layered networks will follow by a simple application of a result from dimension theory. In order to make accessible the results from dimension theory that we shall use, an introduction is given in Section 4 that is geared to give the reader a feeling for the concepts involved. A technical definition is needed to be able to state the results precisely. Two simple examples that attempt to illustrate the concepts introduced in Section 3 and the theoretical results of Section 4 are also included. They show some of the limitations of the theory and also lead to the discovery of certain features of the sigmoid function that may affect the performance of an LNN. These features of the sigmoid functions are stated in the form of Propositions whose proofs, being highly technical, are relegated to the Appendix. In Section 5, we point out some of the interesting conclusions that follow from the results of Section 4. For instance, it will become clear that reducing the dimensionality of the output patterns increases the capacity of the net. A couple of theorems on architectures with maximal capacity are also included in Section 5. The summary and a few conclusions are the content of Section 6.

2. LAYERED NEURAL NETWORKS AND THEIR TRANSFER FUNCTION

An LNN is a network consisting of layers of neurons (processing elements) connected to each other through weighted connections. The output O_i of the i^{th}-neuron is equal to $S(I_i)$, where I_i denotes its input [assumed to be a real number ($I_i \in R$)] and S is a nonlinear function called a *sigmoid function* or *squashing function*. In some cases, S is a *threshold function*. The squashing function S is monotonically increasing, bounded above and below, and usually is differentiable; thus, its graph looks like that in Figure 1. It also could be piecewise linear. Figure 2 shows the graph of a threshold function. Throughout this paper, we shall assume that S is a continuous function mapping R into the interval $I = \{x : -1 \le x \le 1\}$. Let $I^n \equiv \{(x_1, x_2, ..., x_n) : x_i \in I,\ i = 1, 2, ..., n\}$.

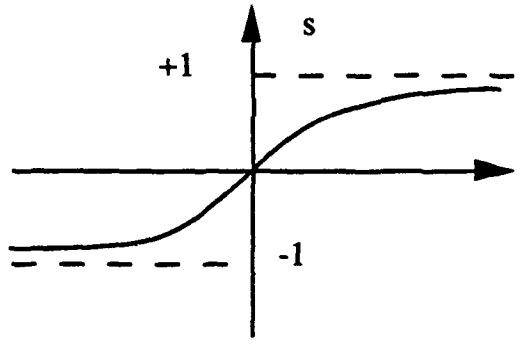

FIGURE 1

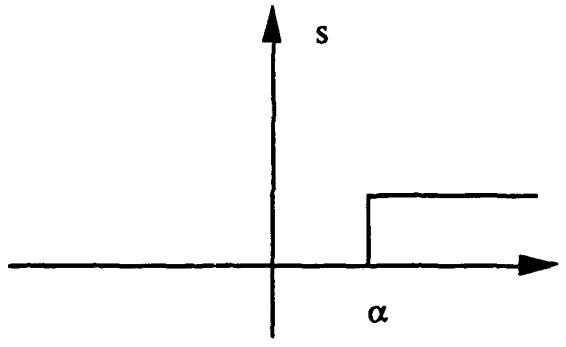

FIGURE 2

The input to a neuron is the weighted sum of the outputs of the neurons in the preceding layer. Thus, between two layers of neurons, layer 1 of size m_1 (a layer with m_1 neurons) and layer 2 of size m_2, we have an $m_2 \times m_1$ - *matrix of weights* W_1. The input to the i^{th}-neuron of layer 2 equals $I_i = \sum_{j=1}^{m_1} w_{ij} O_j$, where O_j denotes the output of the j^{th}-neuron of layer 1, and $W_1 = (w_{ij})$, $(i = 1, 2, 3, ..., m_2;\ j = 1, 2, 3, ..., m_1)$.

Let $\bar{I} = (I_1, I_2, ..., I_{m_2})^T \in R^{m_2}$ denote the vector of inputs to layer 2 and $\bar{O} = (O_1, O_2 ..., O_{m_1})^T \in I^{m_1}$ denote the vector of outputs from layer 1 [$(\cdot)^T$ means transpose of $(\cdot)$]. Then, $\bar{I} = W_1 \bar{O}$ and the output vector of layer 2 is $S_{m_2}(\bar{I})$, where the notation $S_n(x)$ means that $x = (x_1, x_2, ..., x_n)^T \in R^n$ and $S_n(x) = (S(x_1), S(x_2), ..., S(x_n))^T$. Thus,

$$S_{m_2}(W_1 \bar{O}) \tag{2.1}$$

represents the output vector of layer 2 in terms of (a) the output vector $\bar{O}$ of layer 1, (b) the weighting matrix W_1 between layers 1 and 2, and (c) the sigmoid function S_{m_2}.

Formula 2.1 is the basic building block for the general formula (Formula 2.2 below) for multilayered networks. One can think of Formula 2.1

as representing the *Transfer Function* from output of layer 1 to output of layer 2.

Before presenting the general formula for a network with L layers of weights, note that the first layer of neurons (layer 1) has an input that will be the input to the whole network. Denote this input by $\bar{I}_*$. Then, the output $\bar{O}$ of the first layer is simply $\bar{O} = S_{m_1}(\bar{I}_*)$. Now, the general formula for the TF from input $\bar{I}_* \in R^{m_1}$ to output $\bar{O}_* \in I^{m_{L+1}}$ of a neural network with (L + 1) layers of neurons (L layers of weights) of sizes m_i, (i = 1, 2, ..., L + 1) is given by

$$\bar{O}_* = S_{m_{L+1}}(W_L S_{m_L}(W_{L-1} S_{m_{L-1}}(\cdots S_{m_2}(W_1 S_{m_1}(\bar{I}_*))\cdots))). \qquad (2.2)$$

Clearly, the above mapping is a series of compositions of two basic operations: multiplication by a weighting matrix W_k (a linear transformation) followed by the sigmoid function $S_{m_{k+1}}$. So, one can say that each $S_m \circ W_{m-1}$ represents a layer of m neurons.

The notation in Equation 2.2 means the following:

(1) W_k is the k^{th} matrix of weights. These are the weights between layer k of size m_k and layer k+1 of size m_{k+1}, (k = 1, 2, ..., L).

(2) $W_k\, S_{m_k}(\cdot) \in R^{m_{k+1}}$ represents the input to layer k+1, (k = 1, 2, ..., L).

(3) $S_{m_{k+1}}(W_k\, S_{m_k}(\cdot)) \in I^{m_{k+1}}$ represents the output of layer k+1, (k = 1, 2, ..., L).

Note that the subindices of the Ss specify the sizes of the different layers and the dimensions of the matrices in between. Equation 2.2 defines precisely the architecture of the multilayered network.

Remark 2.1 - Each matrix of weights W_i has $m_{i+1} \times m_i$ entries (i = 1, 2, ..., L), and each of these entries can be adjusted independently of the entries

of all the other matrices. Thus, the number M of independent parameters needed to uniquely specify the mapping defined by the right-hand side (RHS) of Equation 2.2 is given by

$$M = \sum_{i=1}^{L} m_{i+1} \times m_i. \tag{2.3}$$

Let us form an M-dimensional vector out of the M weights needed to uniquely specify the RHS of Equation 2.2, call this vector W, and let $T_W: R^{m_1} \rightarrow I^{m_{L+1}}$ represent the mapping $\bar{I}_* \rightarrow \bar{O}_*$ defined by the RHS of Equation 2.2. We have included the vector W in the above notation to remind us that the mapping $\bar{I}_* \rightarrow \bar{O}_*$ depends on $W \in R^M$. With this notation, we can now write Equation 2.2 as

$$\bar{O}_* = T_W(\bar{I}_*) \ , \ (\bar{I}_* \in R^{m_1}). \qquad ////^*$$

Remark 2.2 - Since the sigmoid function S that we are using maps zero into zero (Figure 1), the function S_n also will map the zero vector in R^n into itself, and consequently so will T_W. That is, if θ_n represents the zero vector in R^n, then $T_W(\theta_{m_1}) = \theta_{m_{L+1}}$ no matter what W might be. In some applications, we might want to map θ_{m_1} to a nonzero vector in $R^{m_{L+1}}$. Then it makes sense to introduce a "shift" in the functions S_n. So, instead of using S_n as defined previously, we could define $S_n: R^n \rightarrow I^n$ by

$$S_n(x) = (S(x_1 + \beta_1), S(x_2 + \beta_2), ..., S(x_n + \beta_n))^T, \ \forall x = (x_1, x_2, ..., x_n)^T \in R^n, \tag{2.4}$$

where $\bar{\beta} = (\beta_1, \beta_2, ..., \beta_n)^T$ is a vector of shifts. Now, if the sigmoid functions in Equation 2.2 are shifted as in Equation 2.4, then the mapping so defined has more free parameters. The increase in free parameters is N, where

* The symbol //// indicates the end of a proof, a remark, or an example.

$$N = \sum_{i=1}^{L+1} m_i. \qquad (2.5)$$

Let us form an N-dimensional vector out of the N-shift parameters (call this vector B) and let $T_{W,B}: R^{m_1} \rightarrow I^{m_{L+1}}$ represent the mapping defined by the RHS of Equation 2.2 *when* the sigmoid functions S_{m_i} are shifted by a vector $\bar{\beta}_i \in R^{m_i}$, (i = 1, 2, 3, ..., L+1). Now, the mapping $\bar{I}_* \rightarrow \bar{O}_*$ depends on $W \in R^M$ and also on $B \in R^N$; we can write Equation 2.2 as

$$\bar{O}_* = T_{W,B}(\bar{I}_*), \quad (I_* \in R^{m_1}).$$

Note that if $B = \theta_N$, then $T_{W,B} = T_W$. We shall refer to W as the *weight vector* and B as the *shift vector.* ////

Remark 2.3 - We did not specify exactly how to form the vector W from the entries of the matrices W_i, (i = 1, 2, ..., L), because this will not be relevant for the analysis in Section 4. Only the dimension of W will be relevant. Similarly, we shall not specify how to form the vector B; only its dimension will be of relevance. However, once we decide on a specific way of forming the weight vector W and the shift vector B, then the integer L, the (L + 1)-tuple of indices $\bar{m} \equiv (m_1, m_2, ..., m_{L+1})$, and the vectors W and B *uniquely* describe a particular layered network. So, we may refer to a particular layered network as the *layered network* $(L, \bar{m}, W, B)$ *with Transfer Function* $T_{W,B}: R^{m_1} \rightarrow I^{m_{L+1}}$. ////

3. INTERPRETATION OF $T_{W,B}$

For a *fixed* weight-vector W and a *fixed* shift-vector B, $T_{W,B}$ defines the relation between the inputs and outputs of the layered net. Thus, in this sense, $T_{W,B}$ is a *transfer function.* In this section, we shall interpret $T_{W,B}$ a bit differently, however. We shall think of the *input pattern* $\bar{I}_*$ as a pattern to be

learned or *recognized*; the output $\bar{O}_*$ will be the *desired response* to pattern $\bar{I}_*$. If the weight and shift vectors are such that Equation 2.2 holds, we will say that the network has learned pattern $\bar{I}_*$. So, we will have a collection of input patterns $J = \{\bar{I}_*: * = 1, 2, ..., k\}$ to be learned and a corresponding collection of desired responses $\{\bar{O}_*: * = 1, 2, ..., k\}$. The goal is to determine a set of weights (the learning process) so that Equation 2.2 will hold, at least approximately, for $* = 1, 2, 3, ..., k$. In case the sigmoid functions are shifted, then the goal is to determine a weight-vector W and a shift-vector B so that

$$T_{W,B}(\bar{I}_*) = \bar{O}_*, \quad \text{for } * = 1, 2, 3, ..., k. \tag{3.1}$$

If we think of (W,B) as a point in $R^M \times R^N$, then we can interpret the learning process as a process of finding a point in $R^M \times R^N$ that satisfies (perhaps approximately) the *set of nonlinear equations* defined by Equation 3.1.

Thus, for a given set of input patterns J and a given set of desired responses $\{\bar{O}_*: * = 1, 2, ..., k\}$, Equation 3.1 represents a set of nonlinear equations in the variable $(W,B) \in R^M \times R^N$. We shall call these equations the *Equations of Learning*, because these are the equations that need to be satisfied by (W,B) if the network is to learn the set of patterns J.

Definition 3.1 - The layered network $(L, \bar{m}, W, B)$ with TF function $T_{W,B}$ (see Remark 2.3) is said to have *learned a set of patterns* J *perfectly* if Equation 3.1 is satisfied exactly for $* = 1, 2, ..., k$.

In the next section, we derive an upper bound on the number of patterns that a layered neural network can learn perfectly.

To conclude this section, we will introduce the last set of symbols and notation that will be needed in Section 4. This is done mainly for two reasons: to emphasize (1) the fact that in the equations of learning the *variable* is the

point (W,B) in $R^M \times R^N$, so the notation will explicitly indicate this, and (2) that we are dealing with a mapping F from "(weight × shift)-space" $R^M \times R^N$ into "output-space" $I^{k \cdot m_{L+1}}$.

For a *fixed set of input patterns* $J = \{\bar{I}_* : * = 1, 2, 3, ..., k\}$, define $F: R^M \times R^N \rightarrow I^{k \cdot m_{L+1}}$ by

$$F(W, B) = \begin{bmatrix} T_{W,B}(\bar{I}_1) \\ T_{W,B}(\bar{I}_2) \\ \vdots \\ T_{W,B}(\bar{I}_k) \end{bmatrix}, \quad \forall \ (W, B) \in R^M \times R^N. \tag{3.2}$$

Now we can write the system of Equations 3.1 equivalently as

$$F(W, B) = \begin{bmatrix} \bar{O}_1 \\ \bar{O}_2 \\ \vdots \\ \bar{O}_k \end{bmatrix}.$$

Let $D_J \in I^{k \cdot m_{L+1}}$ denote the vector of desired responses for the set of input patterns J. That is, let

$$D_J = \begin{bmatrix} \bar{O}_1 \\ \bar{O}_2 \\ \vdots \\ \bar{O}_k \end{bmatrix}.$$

Then the equations of learning can succinctly be written as

$$F(W,B) = D_J. \tag{3.3}$$

Remark 3.1 - As the point (W,B) varies through the (weight × shift)-space $R^M \times R^N$, F(W,B) traces the achievable responses. If D_J is an achievable

response, then, of course, Equation 3.3 has a solution. The preimage $F^{-1}(D_J)$ represents the set of all combinations of weight vectors and shift vectors that will solve Equation 3.3. Thus, Equation 3.3 has a solution if—and only if—$F^{-1}(D_J)$ is not empty. (Recall that $F^{-1}(D_J) = \{(W, B) \in R^M \times R^N: F(W, B) = D_J\}$.) ////

Remark 3.2 - The learning process may be interpreted as a process of finding (numerically) approximate solutions to the system of Equation 3.3. Numerical algorithms for solving systems of nonlinear equations such as Newton's Method, Conjugate Gradient, Steepest Descent and others (References 17 and 18) can be used as 'learning algorithms' when applied to Equation 3.3. Thus, a host of new learning algorithms are now at our disposal. Some of these may prove to be faster and/or more efficient than the currently applied learning algorithms such as the Delta-Rule or Back-Propagation-Error (Reference 9). ////

4. CAPACITY OF A LAYERED NEURAL NETWORK

Assume that a set of input patterns J and a vector D_J of desired responses are given and consider the system of Equation 3.3, the equations of learning. If k (the number of patterns to be learned) is too large, then the set of Equation 3.3 will be overdetermined and we may not be able to find a solution (W,B) in $R^M \times R^N$. On the other hand, if k is small enough, there may be an infinite number of solutions to Equation 3.3. Exactly how large k may be and still hope to be able to solve Equation 3.3 is the result we seek. We shall use results from dimension theory to obtain an upper bound on k. Before stating the result formally, we would like to quote a paragraph from the Introduction in Reference 19, p. 7, which expresses the main idea in simple terms:

> "Let $f_i(x_1, ..., x_n)$, $i = 1, ..., m$ be m continuous real valued functions of n real unknowns, or what is the same, m continuous real-valued functions of a point in Euclidean n-space. It is one of the basic facts of analysis that the system of m equations in n unknowns,

$f_i(x_1, ..., x_n) = 0, \ i = 1, 2, ..., m$

has, in general, no solution if $m>n$. The words "in general" may be made precise as follows: by modifying the functions f_i very little one can obtain new continuous functions g_i such that the new system

$g_i(x_1, ..., x_n) = 0, \ i = 1, 2, ..., m$

has no solution. On the other hand, there do exist sets of n equations in n unknowns which are solvable, and which remain solvable after any sufficiently small modification of their left members. This property of Euclidean n-space can be made the basis of a general concept of dimension."

Thus, all we need to do is to count the number of scalar equations that we have in Equation 3.3 and compare that number with the number of unknowns.

Recall that the dimension of the output vectors $\bar{O}_L$ is m_{L+1}; hence, the dimension of D_J is $k \cdot m_{L+1}$ and, therefore, Equation 3.3 is a system of $k \cdot m_{L+1}$ scalar equations. The number of unknowns in Equation 3.3 equals the dimension of W plus the dimension of B; that is, M + N. So, "*in general*," the system of Equation 3.3 has no solution unless $k \cdot m_{L+1} \leq M + N$, or

$$k \leq \frac{1}{m_{L+1}} (M + N), \tag{4.1}$$

where M is given by Equation 2.3 and N is given by Equation 2.5.

Inequality 4.1 is the main result of this work. Interesting consequences that follow from Inequality 4.1 will be discussed in Section 5. The theory needed to give a rigorous justification of Inequality 4.1 is highly technical and it can be found in Reference 19. Here we shall include the minimum amount of theory needed to make this work complete and self-contained.

We should point out that the source of the technical complications is in the statement "the system of m equations in n unknowns $f_i(x_1, ..., x_n) = 0$, (i =

1, ..., m) has, in general, no solution if m > n." This statement is false without the words "in general." Without these words, the resulting statement has many exceptions, some of which are very profound. Therefore, in order to report a correct statement with some degree of generality, one must be very careful of the wording. The following definition will describe precisely what we mean by the words "in general" in the above statement.

Definition 4.1 (Reference 19, p. 74) - Suppose f is a mapping from a space X into a space Y. A point y of f(X) is called an *unstable value* of f if for every positive δ there is a mapping g from X into Y satisfying

(i) $d(f(x), g(x)) < \delta$, for every $x \in X$,

(ii) $g(X) \subset Y - \{y\}$.

Other points of f(X) are called *stable* values of f.

Comments. It is assumed that Y is a metric space and $d(\cdot,\cdot)$ in (i) is the distance function. The notation f(X) represents the image of X under f; that is, $f(X) = \{f(x) : x \in X\}$. If A and B are two sets, then $A - B = \{a \in A : a \text{ is not an element of } B\}$. The symbols "$\subset$" and "$\in$" have the standard set theoretic meaning.

To see the relevance of Definition 4.1 to our problem, consider the equation

$$f(x) = y. \tag{4.2}$$

If $y \in f(X)$, then Equation 4.2 has a solution $x \in X$. However, if y is an unstable value of f, then there are arbitrarily small perturbations of f, like the function g, such that the perturbed equation $g(x) = y$ *has no solution* since $y \notin g(X)$. On the other hand, if y is a stable value of f, then $g(x) = y$ has a solution for *all* sufficiently small perturbations g of f. Clearly, we want to avoid having unstable values on the RHS of Equation 3.3. The next theorem describes cases in which all the values of a function are unstable.

Theorem 4.1 (Reference 19, p. 75) - Let X be a space of dimension *less than* n and f a *mapping* (continuous function) of X into I^n. Then, *all* values of f are unstable.

Fact 4.1 (Reference 19, p. 41) - The dimension of Euclidean n-space R^n is n.

Theorem 4.2 - Let $F : R^M \times R^N \rightarrow I^{k \cdot m_{L+1}}$ be the mapping defined by Equation 3.2. If Inequality 4.1 is violated, then every value of F is unstable.

Proof - Let $X = R^M \times R^N$; then the dimension of X is M + N, since it is homeomorphic to R^{M+N} (Reference 19). If k exceeds the bound in Inequality 4.1, then the dimension of X is less than $k \cdot m_{L+1}$. Since F is continuous, every value of F is unstable by Theorem 4.1. ////

Theorem 4.2 describes the extreme situation in which ***all*** the values of F are unstable. This is certainly an undesirable situation. In this sense, the RHS of Inequality 4.1 is indeed an upper bound on the number k of patterns that the layered net can be expected to learn. Note that it does not guarantee that Equation 3.3 has a solution when k satisfies Inequality 4.1; there may be other conditions to be satisfied by the input-output pairs other than the restriction on the number of pairs imposed by Inequality 4.1. Examples 4.1 and 4.2 below illustrate this point. Because of the architecture of the network, there are certain input-output pairs that are not achievable regardless of the values of the weights or shifts.

Example 4.1 - Consider, for example, the simplest possible case in which L = 1, $m_1 = 1 = m_2$, so that the weight-vector W is a scalar and the shift vector $B = (\beta_1, \beta_2)$ is a 2-vector. The TF is given by

$$T_{W,B}(I) = S(WS(I + \beta_1) + \beta_2), \quad (I \in R^1).$$

Since N = 2, M = 1, and $m_{L+1} = m_2 = 1$, k must be less than or equal to 3. We cannot expect to learn more than three patterns, in general. We shall show

by a simple argument that there are sets of three input-output pairs that are not achievable by this network; hence, they cannot be learned.

Note that $T_{W,B}$ is a monotone function of the input I. This is because it is a composition of two sigmoids which are monotone functions. Thus, if we order the input patterns in an increasing order, say $I_1 < I_2 < I_3$, then only output patterns that are either increasing $O_1 < O_2 < O_3$ or decreasing $O_1 > O_2 > O_3$ are achievable by this network. We can see from this simple example that there may be restrictions on the input-output pairs other than their cardinality. In this case, if the monotonicity condition is not satisfied by the three output patterns, then they are not realizable.

We are going to go a step further and show that for this network and certain sigmoids there are sets of three input-output pairs $\{(I_i, O_i): i = 1, 2, 3\}$ that satisfy the monotonicity condition and are not realizable. This also will serve to illustrate the ideas developed in Section 3 by going through the steps involved in solving the equations of learning for this simple case. For larger networks, of course, one would use numerical methods as pointed out in Remark 3.2.

Suppose then that the pairs to be learned satisfy the monotonicity condition; say $I_1 < I_2 < I_3$ and $O_1 < O_2 < O_3$. We want to find out if there exist a weight W and two shift parameters β_1 and β_2 in R such that the equations of learning are satisfied. That is, so that

$$S(WS(I_i + \beta_1) + \beta_2) = O_i \text{ , for } i = 1, 2, 3. \tag{4.3}$$

We have three equations and three free parameters. Since S is invertible, the first thing we might do to solve this system of equations is to take S^{-1} (the inverse of S) on both sides of Equation 4.3 to obtain

$$WS\,(I_i + \beta_1) + \beta_2 = S^{-1}(O_i)\,, \quad (\,i = 1, 2, 3\,).$$

Next, let i = 1, solve for W, and substitute the expression for W in the other two equations. This leads to

$$W = \frac{S^{-1}(O_1) - \beta_2}{S(I_1 + \beta_1)} \tag{4.4}$$

and

$$\frac{S^{-1}(O_1) - \beta_2}{S(I_1 + \beta_1)} S(I_i + \beta_1) + \beta_2 = S^{-1}(O_i), \quad (i = 2, 3).$$

Now eliminate β_2 from these two equations and obtain

$$\beta_2 = \frac{S^{-1}(O_2)\, S(I_1 + \beta_1) - S^{-1}(O_1)\, S(I_2 + \beta_1)}{S(I_1 + \beta_1) - S(I_2 + \beta_1)} \tag{4.5}$$

and

$$\frac{S^{-1}(O_2)\, S(I_1 + \beta_1) - S^{-1}(O_1)\, S(I_2 + \beta_1)}{S(I_1 + \beta_1) - S(I_2 + \beta_1)} = \frac{S^{-1}(O_3)\, S(I_1 + \beta_1) - S^{-1}(O_1)\, S(I_3 + \beta_1)}{S(I_1 + \beta_1) - S(I_3 + \beta_1)}. \tag{4.6}$$

After eliminating the denominators and simplifying, Equation 4.6 gives

$$a\, S(I_1 + \beta_1) + b\, S(I_2 + \beta_1) + c\, S(I_3 + \beta_1) = 0, \tag{4.7}$$

where $a = S^{-1}(O_3) - S^{-1}(O_2)$, $b = S^{-1}(O_1) - S^{-1}(O_3)$, $c = S^{-1}(O_2) - S^{-1}(O_1)$.

Since $b = -(a + c)$, we can write Equation 4.7 as

$$\frac{S(I_3 + \beta_1) - S(I_2 + \beta_1)}{S(I_2 + \beta_1) - S(I_1 + \beta_1)} = \frac{a}{c}. \tag{4.8}$$

Note that since S is monotonically increasing and $O_1 < O_2 < O_3$, $a > 0$, and $c > 0$, so the ratio $a/c > 0$. Now the question of whether the system of Equations 4.3 has a solution reduces to the question of whether there is a shift parameter β_1 that will solve Equation 4.8. If β_1 exists, then Equation 4.5 gives β_2 and W is given by Equation 4.4 provided $S(I_1 + \beta_1) \neq 0$. Whether or not Equation 4.8 has a

solution depends on the specific sigmoid function S. Up to this point, for the sake of generality, the sigmoid function S has been left unspecified. The analysis of the previous section holds independently of the particular details of the specific sigmoid function employed as long as it is continuous. Thus, the results we have so far are valid *for all sigmoids*, which form a large class of functions. In fact, they are valid for any continuous function S. However, the question of whether Equation 4.8 has a solution clearly depends on the specific sigmoid used. If the sigmoid has compact range—that is, if it saturates above and below so that the sigmoid is a constant for large and small values of its argument—then Equation 4.8 always has a solution (Proposition 4.1). Otherwise, it may not have a solution (Proposition 4.2). Thus, the shape of the sigmoid is indeed relevant at this point. ////

Definition 4.2 - By a *sigmoid that saturates* we shall mean a continuous, nondecreasing function S: $R^1 \rightarrow [-1, 1]$ that is onto and is strictly increasing on $S^{-1}((-1, 1))$. (Recall that $S^{-1}((-1, 1)) = \{x \in R^1 : S(x) \in (-1, 1)\}$.)

Proposition 4.1 - If S is a sigmoid that saturates, then Equation 4.8 always has a solution.

We can show (Proposition 4.2 below) that if S is an inverse tangent, $S(t) = \frac{2}{\Pi} \tan^{-1}(t)$, then there are inputs I_1, I_2, I_3 so close to each other that the ratio in the LHS of Equation 4.8 is greater than some positive number η *for all* $\beta_1 \in R$. Thus, if the outputs O_i are such that $a/c < \eta$, then Equation 4.8 has no solution.

Proposition 4.2 - Let $S(t) = \frac{2}{\Pi} \tan^{-1}(t)$, $(t \in R)$. Let $I_k = k - 1$ for $k = 1, 2, 3$, and let

$$\gamma(\beta) = \frac{S(I_3 + \beta) - S(I_2 + \beta)}{S(I_2 + \beta) - S(I_1 + \beta)}, \quad (\beta \in R).$$

There exists a positive number η such that $\gamma(\beta) \geq \eta$ for all $\beta \in R$.

These two propositions are proved in the Appendix.

Remark 4.1 - The significance of Proposition 4.2 is only theoretical, since any hardware implementation of the sigmoid would result in a sigmoid that saturates above and below, achieving a maximum constant value for large arguments and a minimum constant value for large negative arguments. Proposition 4.1 says that this saturation is a *desirable* feature of the sigmoid. ////

Example 4.2 - Consider the layered network defined by $(L, \bar{m}, W, B) \equiv (2, (1, 2, 1), \begin{bmatrix} W_1 = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\ W_2 = [w_3 w_4] \end{bmatrix}, \theta_4)$ with TF $T_W(I) = S\left([w_3 w_4]\, S_2\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} I \right) \right) =$ $S(w_3\, S(w_1 I) + w_4 S(w_2 I))$, $(I \in R)$ (see Remark 2.3).

Note that the shift vector B is zero in this example and, since the sigmoid is an *odd function* (i.e., S(-I) = -S(I) for all I), it is easy to see that T_W is also an odd function. Thus, the input-output pairs of this network cannot be all that arbitrary. Regardless of what the weights may be, there are two constraints that must be satisfied by the input-output pairs of this system due to the fact that T_W is an odd function:

(i) $T_W(0) = 0$ and

(ii) $T_W(-I) = -T_W(I)$.

For this network, Inequality 4.1 gives $k \leq 4$. Hence we cannot expect this network to learn more than four arbitrary input-output pairs; moreover, if some pair violates (i) or (ii), then the network will not learn it. So, again we see that Inequality 4.1 gives an upper bound on k; however, there is no guarantee that Equation 3.3 has a solution when k satisfies Inequality 4.1.

We can use this example to illustrate how the opposite situation also can occur. This is the fortuitous situation in which k exceeds the bound in Inequality 4.1 yet a solution to Equation 3.3 exists.

Suppose that the network of Example 4.2 has learned the four input-output pairs (I_i, O_i), $i = 1, 2, 3, 4$, where none of the inputs I_i is zero and no two of them satisfy $I_i = -I_j$ for $i \neq j$. Then, under these conditions, we could "add" five more input-output pairs to the list of pairs to be learned. These are the negatives of the four pairs above, $(-I_i, -O_i)$, $i = 1, 2, 3, 4$, and the pair (0, 0). All nine pairs are realizable, so it would seem that we have violated Inequality 4.1, since now $k = 9 > 4$. However, this is just a fortuitous situation; the extra four pairs $(-I_i, -O_i)$, $i = 1, 2, 3, 4$, do not impose any new conditions on the weights. They are "learned" as a by-product of learning the first four pairs (I_i, O_i), $i = 1, 2, 3, 4$. The pair (0,0) results from not having shifts in the sigmoids (see Remark 2.2). ////

5. APPLICATIONS

Inequality 4.1 together with Theorem 4.2 represent the main result of this work. An attempt was made to illustrate the meaning of this result. This was the purpose of Examples 4.1 and 4.2. It is hoped that the comments, remarks, and examples in Section 4 are sufficient to justify calling the quantity on the RHS of Inequality 4.1 "the capacity" of the layered net.

By Equations 2.3 and 2.5, Inequality 4.1 says

$$k \leq \frac{1}{m_{L+1}} \left[\sum_{i=1}^{L} m_{i+1} \times m_i + \sum_{i=1}^{L+1} m_i \right]. \tag{5.1}$$

Definition 5.1 - If a LNN has L layers of weights ($L \geq 1$) and ($L + 1$) layers of neurons with m_i neurons in the i^{th}-layer, $m_i \geq 1$, $i = 1, 2, 3, \ldots, (L + 1)$, then the $(L + 1)$-tuple $\bar{m} \equiv (m_1, m_2, \ldots, m_{L+1})$ will be called the *architecture* of the LNN.

Definition 5.2 - If a LNN has architecture $\bar{m} = (m_1, m_2, ..., m_{L+1})$, then the RHS of Inequality 5.1 will be called the *capacity* of the LNN and will be denoted by $C(\bar{m})$. Thus,

$$C(\bar{m}) = \frac{1}{m_{L+1}}\left[\sum_{i=1}^{L} m_i \times m_{i+1} + \sum_{i=1}^{L+1} m_i\right], \quad (\bar{m} \in N^{L+1}). \tag{5.2}$$

Here, N^{L+1} represents the collection of (L + 1)-tuples of positive integers.

In this terminology, Theorem 4.2 says that if the number of input patterns exceeds the capacity of the LNN, then every value of the mapping F defined by Equation 3.2 is unstable.

Now that the notions of "architecture" and "capacity" of an LNN have been defined, we shall derive a few simple results that follow from Definition 5.2. The first one, which is perhaps counterintuitive, is the fact that the capacity of an LNN *decreases* as the number of output neurons *increases*. This becomes evident after we rearrange terms and write $C(\bar{m})$ as

$$C(\bar{m}) = 1 + m_L + \frac{1}{m_{L+1}}\left[\sum_{i=1}^{L-1} m_{i+1} \times m_i + \sum_{i=1}^{L} m_i\right]. \tag{5.3}$$

Clearly, $C(\bar{m})$ is a decreasing function of the number of output neurons m_{L+1}. Thus, increasing m_{L+1} can only decrease $C(\bar{m})$. In fact, if the total number of neurons is fixed, then m_{L+1} can increase only at the expense of decreasing some of the m_j $(j \leq L)$, which clearly would reduce $C(\bar{m})$ even further. So, we have discovered the following practical result.

Theorem 5.1 - (a) Reducing the number of output neurons while keeping the total number of neurons fixed increases the capacity of the layered network. (b) Reducing the number of output neurons while keeping the number of neurons in all the other layers fixed increases the capacity of the layered network.

Note that in part (b) the total number of neurons actually decreases by decreasing the number of output neurons, yet the capacity still increases.

In practice, one may not always be able to choose m_{L+1}, the number of neurons in the output layer. However, in those applications where one has the freedom to choose m_{L+1}, Theorem 5.1 suggests that it be chosen as small as possible.

We conclude this section with a series of results on maximizing architectures: that is, architectures that will maximize the capacity $C(\bar{m})$ of an LNN when the total number of neurons is fixed and with a fixed number of layers of weights.

The total number of neurons of the LNN will be denoted by N as in Equation 2.5 and, as usual, L will denote the number of layers of weights. For a given L, the architecture that maximizes the capacity of the net will be denoted by $\bar{m}_L^*$, and $C_L^* = C(\bar{m}_L^*)$ is the maximal capacity of the net.

Theorem 5.2 - Let $m_1 = \alpha$, $m_{L+1} = \beta$, and $N = \sum_{i=1}^{L+1} m_i$ be fixed positive integers.

(a) If $L = 2$ and $N > \alpha + \beta$, then $\bar{m}_2^* = (\alpha, N - \alpha - \beta, \beta)$ and $C_2^* = \frac{1}{\beta}\left[(1 + \alpha + \beta)N - (\alpha + \beta)^2\right]$.

(b) If $L = 3$, $N > 2\beta$, $N > 2\alpha$, and N is even, then $\bar{m}_3^* = (\alpha, N/2 - \beta, N/2 - \alpha, \beta)$ and $C_3^* = \frac{1}{\beta}\left[N + N^2/4 - \alpha\beta\right]$.

(c) If $\alpha > \beta$, $L = 4$, $(N - \beta)$ is even, and $(N - \beta) > 2\alpha$, then $\bar{m}_4^* = \left(\alpha, \frac{1}{2}(N - \beta) - 1, \frac{1}{2}(N - \beta) - \alpha, 1, \beta\right)$ and $C_4^* = \frac{1}{\beta}\left[N + (N - \beta)^2/4 + (\beta - \alpha)\right]$.

(d) If $\alpha < \beta$, $L = 4$, $(N - \alpha)$ is even, and $(N - \alpha) > 2\beta$, then $\bar{m}_4^* = \left(\alpha, 1, \frac{1}{2}(N - \alpha) - \beta, \frac{1}{2}(N - \alpha) - 1, \beta\right)$ and $C_4^* = \frac{1}{\beta}\left[N + (N - \alpha)^2/4 + (\alpha - \beta)\right]$.

(e) If $\alpha = \beta$, $L = 4$, $(N - \alpha)$ is even, and $N - 3\alpha \geq 2$, then $\bar{m}_4^* = \left(\alpha, \frac{1}{2}(N - \alpha) - m_4, \frac{1}{2}(N - 3\alpha), m_4, \alpha\right)$, with $1 \leq m_4 \leq \frac{1}{2}(N - \alpha) - 1$, and $C_4^* = \frac{1}{\beta}\left[N + (N - \alpha)^2/4\right]$.

(f) Under the hypothesis of (b) and (c), or (d), or (e), $C_3^* > C_4^*$.

(g) $C_2^* \geq C_3^*$ if, and only if, $4\alpha\beta \geq [2(a + \beta) - N]^2$.

Proof - See Appendix.

Remark 5.1 - If m_1 and m_3 are required to equal α and β, respectively, then, when $L = 2$, there is only one choice for $\bar{m}$; thus, $\bar{m}_2^*$ is necessarily this unique choice. In this sense, the result of part (a) is trivial. However, it is included so that C_2^* can be compared with C_3^* and C_4^*. Part (f) suggests that L be chosen no larger than 3 in situations where it is important to achieve maximal capacity. The results of parts (c) and (d) are interesting, for they show that when $L = 4$ and $\alpha \neq \beta$, maximal capacity requires an architecture with a layer that has only one neuron. This phenomenon persists when $L > 4$. Finally, we can see from part (g) that if N is "small" compared with $2(\alpha + \beta)$—"small" being defined by the inequality in (g)--then $L = 2$ gives a larger capacity than $L = 3$. ////

As Theorem 5.1 shows, in order to maximize the capacity of an LNN, m_{L+1} should be selected as small as possible. In the next theorem, we are going to set $m_{L+1} = 1$ and let m_1 be free, with the total number of neurons fixed. The results that follow will complement those of Theorem 5.2, where m_1 was fixed at a given value α and m_{L+1} was fixed at β. This will tell us what the optimal value of α would be when $\beta = 1$.

Theorem 5.3 - Let $m_{L+1} = 1$ and $N = \sum_{i=1}^{L+1} m_i$ be fixed.

(a) If $L = 2$ and N is even, then $\bar{m}_2^* = \left(\frac{1}{2}N - 1, \frac{1}{2}N, 1\right)$ and $C_2^* = N + N^2/4$.

(b) If $L = 3$ and N is even, then $\bar{m}_3^* = \left(1, \frac{1}{2}N - 1, \frac{1}{2}N - 1, 1\right)$ and $C_3^* = N + \frac{1}{4}N^2 - 1$.

(c) If $L = 4$ and N is odd, then $\bar{m}_4^* = \left(1, \frac{1}{2}(N-1) - m_4, \frac{1}{2}(N-3), m_4, 1\right)$, with $1 \le m_4 \le \frac{1}{2}(N-3)$ and $C_4^* = \frac{1}{4}(N+1)^2$.

Proof - See Appendix.

Note that under the conditions of Theorem 5.3, we now have $C_2^* > C_3^*$ and the maximal capacity decreases as L increases. Also, note that maximal capacity calls for $m_1 = 1$ when $L = 3$ or 4. This phenomenon persists when $L > 4$.

6. SUMMARY AND CONCLUSIONS

The essential components and the architecture of a general (feed-forward) LNN were defined. The equations that must be satisfied by the weights and shifts of the network in order to learn a set of input-output pairs were derived. These form a set of nonlinear equations that were called the equations of learning. If the set of pairs to be learned is known a priori, then one can (in principle) write down these equations and interpret the learning process as a process of solving (numerically) the equations of learning. Thus, a host of new learning algorithms are now at our disposal; namely, all of the known algorithms for solving systems of nonlinear equations (References 17 and 18), which, in this setting, can be interpreted as learning algorithms. Some of these may prove to be faster and/or more efficient than any of the currently applied learning algorithms, such as the Delta-Rule or Back-Propagation-Error (Reference 9), which is one of the most popular methods.

Whether one can exploit the special structure of the equations of learning to improve or adapt some of the existing algorithms for solving systems of equations, so that these algorithms become useful as learning algorithms that might compete against existing learning algorithms, is a topic of further research which will not be addressed here. It is left as an open problem.

By counting the number of degrees of freedom of a layered net (which is equal to the dimension of the (weight × shift)-space of the net) and bounding the number of equations in the equations of learning by this number, an upper bound on the number of distinct input-output pairs that the net can learn was obtained. Example 4.1 together with Proposition 4.1 show that the upper limit can be achieved making the bound sharp. Two examples to illustrate the result of Theorem 4.2 and its limitations where included.

While analyzing Example 4.1, it was discovered that one can always solve Equation 4.8 if the sigmoid function saturates. It was discovered also that if the sigmoid approaches its asymptotic values too slowly (like an arctangent for example), then Equation 4.8 may not have a solution if the inputs are too close to each other. Thus, saturation is a desirable feature of the sigmoid. This result is significant, because the simple net of Example 4.1 can be thought of as a building block for larger nets. Whenever a net has two or more layers of neurons with forward connections, one has the simple net of Example 4.1 embedded in it.

Finally, in Section 5, we defined the concepts of architecture and capacity of an LNN and gave a few results on architectures with maximal capacity.

Appendix

PROOF OF PROPOSITIONS 4.1 AND 4.2 AND THEOREMS 5.2 AND 5.3.

Observation A.1 - If S: $R^1 \rightarrow [-1, 1]$ is a sigmoid that saturates, then there exist two points t_1 and t_2 such that $t_1 < t_2$, $(-\infty, t_1] = S^{-1}(\{-1\})$, $[t_2, \infty) = S^{-1}(\{+1\})$, and on $[t_1, t_2]$ S is a strictly increasing function with range [-1, 1].

Proposition 4.1 - Let $I_1 < I_2 < I_3$ be three real numbers and α any non-negative number. If S is a sigmoid that saturates, then there exists $\beta \in R$ such that

$$\frac{S(I_3 + \beta) - S(I_2 + \beta)}{S(I_2 + \beta) - S(I_1 + \beta)} = \alpha. \tag{A.1}$$

Proof - For each $\beta \in R$, let $g_1(\beta) \equiv S(I_3 + \beta) - S(I_2 + \beta)$ and $g_2(\beta) \equiv S(I_2 + \beta) - S(I_1 + \beta)$. Since S is strictly increasing on $[t_1, t_2]$ and $I_1 < I_2$, we conclude that $g_2(\beta) > 0$ for every $\beta \in (t_1 - I_2, t_2 - I_2]$.

If $\mathcal{D} \equiv (t_1 - I_2, t_2 - I_2]$ and $f(\beta) \equiv \dfrac{g_1(\beta)}{g_2(\beta)}$ for every $\beta \in \mathcal{D}$, then f is the ratio of two continuous functions with a denominator that never vanishes; thus, f is continuous on $\mathcal{D}$. Next, we shall show that the range of f is the infinite interval $[0, \infty)$. Since f is continuous, this can be accomplished by showing that $f(t_2 - I_2) = 0$ and

$$\lim_{\beta \rightarrow t_1 - I_2} f(\beta) = \infty. \tag{A.2}$$

Recall that $S(\beta) \equiv 1$ for $\beta \geq t_2$ and, since $I_3 > I_2$, $g_1(t_2 - I_2) = 0$. Hence, $f(t_2 - I_2) = 0$. To see that Equation A.2 holds, let $b \equiv g_1(t_1 - I_2)$. Since $I_3 > I_2$ and S is strictly increasing on $[t_1, t_2]$, $b > 0$. (Note that if $I_3 - I_2 + t_1 > t_2$, then $b = 2$.)

Next, since g_1 is continuous, there exists $\varepsilon > 0$ such that $g_1(\beta) > b/2$ for all $\beta \in (t_1 - I_2, t_1 - I_2 + \varepsilon)$. Therefore,

$$f(\beta) > \frac{b}{2g_2(\beta)} \quad \text{for all } \beta \in (t_1 - I_2, t_1 - I_2 + \varepsilon). \tag{A.3}$$

Now, since $g_2(t_1 - I_2) = 0$ and g_2 is continuous, $g_2(\beta)$ becomes arbitrarily small as $\beta \to (t_1 - I_2)$. Consequently, Inequality A.3 implies Equation A.2.

We have shown that the LHS of Equation A.1, represented by $f(\beta)$, can attain any nonnegative value as β varies through the interval $\mathcal{D}$; thus, whenever $\alpha \geq 0$, Equation A.1 has a solution with $\beta \in \mathcal{D}$. ////

Proposition 4.2 - Let $S(t) = \frac{2}{\Pi} \tan^{-1}(t)$, $(t \in R)$. Let $I_k = k - 1$ for $k = 1, 2, 3$ and let

$$\gamma(\beta) \equiv \frac{S(I_3 + \beta) - S(I_2 + \beta)}{S(I_2 + \beta) - S(I_1 + \beta)}, \quad (\beta \in R). \tag{A.4}$$

There exists a positive number η such that $\gamma(\beta) \geq \eta$ for all $\beta \in R$.

Proof - Let S' denote the derivative of S. The following function of two variables will be instrumental in the proof.

$$f(x, y) \equiv \frac{S'(x + y)}{S'(x)}, \quad x \in R, \quad y \in [0, 2].$$

Since $S'(x) > 0$ for all x, $f(x, y) > 0$ for all $x \in R$ and $y \in [0, 2]$. Furthermore, f is a continuous function. Thus, if we restrict f to a compact rectangle of the form $[a, b] \times [0, 2]$, where $a < b$, f will attain a minimum value $\lambda > 0$. We shall use this fact later.

By the Mean Value Theorem, for each $\beta \in R$, we can find two numbers, $\alpha_1(\beta)$ and $\alpha_2(\beta)$, such that

$$\begin{cases} \beta < \alpha_1(\beta) < 1 + \beta \\ 1 + \beta < \alpha_2(\beta) < 2 + \beta \quad \text{and} \end{cases} \tag{A.5}$$

$$\begin{cases} S(2 + \beta) - S(1 + \beta) = S'(\alpha_2(\beta)) \\ S(1 + \beta) - S(\beta) = S'(\alpha_1(\beta)) . \end{cases} \tag{A.6}$$

Let $T(\beta) \equiv \alpha_2(\beta) - \alpha_1(\beta)$, so that $\alpha_2(\beta) = \alpha_1(\beta) + T(\beta)$. Then, by Inequalities A.5,

$$T(\beta) \in (0, 2), \ \forall \beta \in R. \tag{A.7}$$

Comparing Equations A.4 and A.6 we get

$$\gamma(\beta) = \frac{S'(\alpha_1(\beta) + T(\beta))}{S'(\alpha_1(\beta))}, \ (\beta \in R). \tag{A.8}$$

Since $S'(t) = \frac{2}{\Pi} \frac{1}{1 + t^2}$, $(t \in R)$, a quick calculation using Equation A.8 shows that $\gamma(\beta) = \frac{1 + \alpha_1^2(\beta)}{1 + [\alpha_1(\beta) + T(\beta)]^2}$, $(\beta \in R)$, which, for all $T(\beta) \in [0, 2]$, approaches 1 as $|\alpha_1(\beta)| \to \infty$. Therefore, since $\beta < \alpha_1(\beta) < \beta + 1$, there exists $\bar{\beta} > 0$ such that

$$\gamma(\beta) > \frac{1}{2}, \text{ for } |\beta| > \bar{\beta}. \tag{A.9}$$

Next, let $\lambda > 0$ be the minimum value of $f(x, y)$ for $x \in [-\bar{\beta}, \bar{\beta} + 1]$ and $y \in [0, 2]$. By the choice of λ, Inequality A.5, Statement A.7, and Equation A.8 we conclude

$$\gamma(\beta) \geq \lambda \ \text{ for } |\beta| \leq \bar{\beta}. \tag{A.10}$$

Now, if $\eta \equiv \min\{\lambda, \frac{1}{2}\}$, then Inequalities A.9 and A.10 imply that $\gamma(\beta) \geq \eta$ for all $\beta \in R$. ////

Proof of Theorem 5.2 - The expressions for C_2^*, C_3^*, and C_4^* can be obtained from Equation 5.2 and $\bar{m}_2^*$, $\bar{m}_3^*$, and $\bar{m}_4^*$, respectively, using elementary algebra. This is also true of $\bar{m}_2^*$ (see Remark 5.1).

Since m_1, m_{L+1}, and N are fixed and $N = \sum_{i=1}^{L+1} m_i$, the capacity $C(\bar{m})$ can be expressed as a function of L - 2 variables when $L \geq 3$. For each $L \geq 3$ and $(m_3, m_4, ..., m_L) \in R^{L-2}$, let $f_L(m_3, m_4, ..., m_L) \equiv \beta \cdot C(\alpha, N - \alpha - \beta - \sum_{i=3}^{L} m_i, m_3, ..., m_L, \beta)$. Note that this amounts to setting $m_2 = N - \alpha - \beta - \sum_{i=3}^{L} m_i$. It suffices to show that f_L achieves its global maximum when m_i equals the i^{th}-component of $\bar{m}_L^*$ (i = 3, 4, ..., L; L = 3, 4).

If L = 3, then $f_3(m_3) = N + [\alpha + m_3][N - \alpha - \beta - m_3] + \beta m_3$, $(m_3 \in R)$. Let $m_3^* = \frac{N}{2} - \alpha$, then a simple calculation gives $f_3(m_3^* + x) = \beta C_3^* - x^2$, $\forall x \in R$. Thus, $f_3(m_3^* + x) \leq f_3(m_3^*) = \beta C_3^*$, $\forall x \in R$. This implies that $f_3(m_3^*)$ is the global maximum of f_3 and (b) has been established.

If L = 4, then $f_4(m_3, m_4) = N + [\alpha + m_3][N - \alpha - \beta - m_3 - m_4] + m_3 m_4 + \beta m_4$, $(m_3, m_4) \in R^2$. Let $m_3^* = \frac{1}{2}(N - \beta) - \alpha$, $m_4^* = 1$ and assume $\alpha > \beta$. A simple calculation gives

$$f_4(m_3^* + x, m_4^* + y) = \beta C_4^* - [x^2 + (\alpha - \beta)y], \quad \forall (x, y) \in R^2.$$

Since $x^2 + (\alpha - \beta)y \geq 0$ for all $x \in R$ and all $y \geq 0$, we have

$$f_4(m_3^* + x, m_4^* + y) \leq f_4(m_3^*, m_4^*) = \beta C_4^*, \quad \forall x \in R, y \geq 0. \tag{A.11}$$

Note that since $m_4^* = 1$, it cannot be perturbed by $y < 0$ in our setting. It follows from Inequality A.11 that $f_4(m_3^*, m_4^*)$ is the global maximum of f_4, which establishes (c).

Since $m_5 C(m_1, m_2, m_3, m_4, m_5) = m_1 C(m_5, m_4, m_3, m_2, m_1)$, (d) follows from (c) by interchanging the roles of α and β.

If $\alpha = \beta$ and $m_3^* = \frac{1}{2}(N - 3\alpha)$, then

$$f_4(m_3^* + x, y) = N + (N - \alpha)^2/4 - x^2 = \beta C_4^* - x^2, \ \forall (x, y) \in R^2.$$

This shows that f_4 achieves its global maximum on the set of points $\{(m_3^*, y) : y \in R\}$. However, in our setting y must be restricted to the allowable values of m_4, which are $1 \le m_4 \le \frac{1}{2}(N - \alpha) - 1$. The lower bound on m_4 is clear. The upper bound is determined by the requirement that $m_2 \ge 1$. Since $m_2 = N - 2\alpha - m_3 - m_4$, when $m_3 = m_3^*$, we have $m_2 = \frac{1}{2}(N - \alpha) - m_4$. Hence, $m_2 \ge 1$ implies $m_4 \le \frac{1}{2}(N - \alpha) - 1$. Now (e) has been established.

Under the hypothesis of part (c), we have $2\left[\frac{\alpha}{\beta} - 1\right] > 0$ and $(N - 2\alpha) - \frac{1}{2}\beta > 0$, which imply the following series of inequalities:

$$0 < \frac{1}{2}\beta\left[N - \frac{1}{2}\beta - 2\alpha + 2\frac{\alpha}{\beta} - 2\right]$$

$$\Rightarrow \quad -\frac{1}{2}\beta N + \frac{1}{4}\beta^2 - \alpha + \beta < -\alpha\beta$$

$$\Rightarrow \quad \frac{1}{4}\left[N^2 - 2\beta N + \beta^2\right] + (\beta - \alpha) < \frac{1}{4}N^2 - \alpha\beta$$

$$\Rightarrow \quad N + (N - \beta)^2/4 + (\beta - \alpha) < N + \frac{1}{4}N^2 - \alpha\beta.$$

Thus, when $\alpha > \beta$, $C_4^* < C_3^*$.

Under the hypothesis of part (d), we have $2\left(\frac{\beta}{\alpha} - 1\right) > 0$ and $(N - 2\beta) - \frac{1}{2}\alpha > 0$. These give (as above, by interchanging α and β)

$$N + (N - \alpha)^2/4 + (\alpha - \beta) < N + \frac{1}{4}N^2 - \alpha\beta.$$

Thus, when $\alpha < \beta$, $C_4^* < C_3^*$.

Under the hypothesis of part (e), we have $2N - 6\alpha \geq 4$; hence,

$$2N - 5\alpha \geq 4 + \alpha > 0 \Rightarrow -4\alpha > -2N + \alpha \Rightarrow -4\alpha^2 > -2\alpha N + \alpha^2$$

$$\Rightarrow N^2 - 4\alpha^2 > (N - \alpha)^2 \Rightarrow N + \frac{1}{4}N^2 - \alpha^2 > N + (N - \alpha)^2/4.$$

Thus, when $\alpha = \beta$, $C_4^* < C_3^*$. This establishes (f).

Finally, we have

$$4\alpha\beta \geq [2(\alpha + \beta) - N]^2$$

$$\Leftrightarrow \quad \alpha\beta \geq (\alpha + \beta)^2 - (\alpha + \beta)N + \frac{N^2}{4}$$

$$\Leftrightarrow \quad -(\alpha + \beta)^2 + (\alpha + \beta)N \geq \frac{N^2}{4} - \alpha\beta$$

$$\Leftrightarrow \quad N + (\alpha + \beta)N - (\alpha + \beta)^2 \geq N + \frac{N^2}{4} - \alpha\beta$$

$$\Leftrightarrow \quad C_2^* \geq C_3^*.$$

This completes the proof of Theorem 5.2. ////

Proof of Theorem 5.3 - We shall use the same technique used in the proof of Theorem 5.2. The expressions for C_2^*, C_3^*, and C_4^* can be obtained from Equation 5.2 and $\bar{m}_2^*$, $\bar{m}_3^*$, and $\bar{m}_4^*$, respectively, using elementary algebra.

For each $L \geq 2$ and $(m_2, m_3, ..., m_L) \in R^{L-1}$, let

$$f_L(m_2, m_3, ..., m_L) \equiv C(N - 1 - \sum_{i=2}^{L} m_i, m_2, m_3, ..., m_L, 1).$$

Note that this amounts to setting

$$m_1 = N - 1 - \sum_{i=2}^{L} m_i \ , \ (L \geq 2). \qquad (A.12)$$

It will be shown that f_L achieves its global maximum when m_i equals the i^{th}-component of $\bar{m}_L^*$ (i = 2, 3, ..., L; L = 2, 3, 4).

If L = 2, then $f_2(m_2) = N + [N - 1 - m_2]\, m_2 + m_2 = N + [N - m_2]\, m_2$. Let $m_2^* = \frac{1}{2}N$, then $f_2(m_2^* + x) = N + \frac{N^2}{4} - x^2 = f_2(m_2^*) - x^2$ $(x \in R)$. Thus, $f_2(m_2^* + x) \leq f_2(m_2^*) = C_2^*, \forall x \in R$. This gives (a).

If L = 3, then $f_3(m_2, m_3) = N + [N - 1 - m_2]m_2 + m_3$. Let $m_2^* = \frac{1}{2}N - 1 = m_3^*$, then $f_3(m_2^* + x, m_3^* + y) = C_3^* - x^2 + (x + y)$. Since $m_1 \geq 1$, Equation A.12 implies that $x + y \leq 0$. Therefore, $f_3(m_2^* + x, m_3^* + y) \leq f_3(m_2^*, m_3^*) = C_3^*$ whenever $x + y \leq 0$. This shows that C_3^* is the global maximum of C_3 under the constraints of Equation A.12 and $m_1 \geq 1$. This gives (b).

If L = 4, $m_2^* = \frac{1}{2}(N - 1) - m_4$, $m_3^* = \frac{1}{2}(N - 3)$, and m_4 is arbitrary for the moment, then

$$f_4(m_2^* + x, m_3^* + y, m_4) = \frac{1}{4}[N + 1]^2 - x^2 + m_4(x + y) = C_4^* - x^2 + m_4(x + y).$$

As in case (b), $m_1 \geq 1$ and Equation A.12 implies $x + y \leq 0$. Therefore, $f_4(m_2^* + x, m_3^* + y, m_4) \leq f_4(m_2^*, m_3^*, m_4) = C_4^*$ whenever $(x + y) \leq 0$, and for all $m_4 \in R$. This shows that C_4^* is the global maximum of C_4 under the constraints of Equation A.12 and $m_1 \geq 1$, independently of the value of m_4. However, in our setting, we must have $1 \leq m_4 \leq \frac{1}{2}(N - 3)$. The lower bound on m_4 is clear. The upper bound is obtained by noticing that $m_2^* = \frac{1}{2}(N - 1) - m_4 \geq 1$. This completes the proof of Theorem 5.3. ////

REFERENCES

1. J. J. Hopfield. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. Nat. Acad. Sci. USA*, Vol. 79 (1982), pp. 2554-58.

2. J. J. Hopfield and D. W. Tank. "Neural Computation of Decisions in Optimization Problems," *Biol. Cybern.*, Vol. 52 (1985), pp. 141-52.

3. G. A. Carpenter and S. Grossberg. "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Comput. Vision Graphics and Image Process*, Vol. 37 (1987), pp. 54-115.

4. _____. "ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns," *Proceedings of the IEEE First International Conference on Neural Networks*. San Diego, Calif. 21-24 June (1987). Pp. II-727-35.

5. S. Grossberg. *Studies of Mind and Brain.* Boston, Reidel, 1982.

6. _____. "Adaptive Pattern Classification and Universal Recoding: II. Feedback. Expectation, Olfaction, Illusions," *Biol. Cybern., Vol. 23* (1976), pp. 187-202.

7. B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Englewood Cliffs, N.J., Prentice Hall, 1985.

8. B. Widrow and R. Winter. "Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition," *Comput.* (March 1988), pp. 25-39.

9. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation," *ICS Report 8506*. San Diego, Calif., University of California, 1985.

10. D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing*, Vols. 1 and 2. Cambridge, Mass., MIT Press, 1986.

11. Y. S. Abu-Mostafa and J. St. Jacques. "Information Capacity of the Hopfield Model," *IEEE Trans. Inf. Theory*, Vol. IT-31 (1985), pp. 461-64.

12. J. F. Fontanari and R. Köberle. "Information Storage and Retrieval in Synchronous Neural Networks," *Phys. Rev. A*, Vol. 36, No. 5 (September 1987), pp. 2475-77.

13. M. R. B. Forshaw. "Pattern Storage and Associative Memory in Quasi-Neural Networks," *Pattern Recognition Lett. Vol. 4* (1986), pp. 427-31.

14. W. A. Little and G. L. Shaw. "Analytic Study of the Memory Storage Capacity of a Neural Network," *Math. Biosci.*, Vol. 39 (1978), pp. 281-90.

15. R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh. "The Capacity of the Hopfield Associative Memory," *IEEE Trans. Inf. Theory*, Vol. IT-33, No. 4 (July 1987), pp. 461-82.

16. S. 'Olafsson and Y. S. Abu-Mostafa. "The Capacity of Multilevel Threshold Functions," *IEEE Trans. Pattern Analy. Machine Intell.*, Vol. 10, No. 2 (March 1988), pp. 277-81.

17. K. E. Atkinson. *An Introduction to Numerical Analysis.* New York, John Wiley & Sons, 1978.

18. J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, N.J., Prentice-Hall, 1983.

19. W. Hurewicz and H. Wallman. *Dimension Theory.* Princeton, N.J., University Press, 1941.